Correction of Medical Handwriting OCR Based on Semantic Similarity*

Bartosz Broda[1] and Maciej Piasecki[1]

Institute of Applied Informatics, Wrocław University of Technology, Poland
{bartosz.broda,maciej.piasecki}@pwr.wroc.pl

Abstract. In the paper a method of the correction of handwriting Optical Character Recognition (OCR) based on the semantic similarity is presented. Different versions of the extraction of semantic similarity measures from a corpus are analysed, with the best results achieved for the combination of the text window context and Rank Weight Function. An algorithm of the word sequence selection with the high internal similarity is proposed. The method was trained and applied to a corpus of real medical documents written in Polish.

Keywords: semantic similarity, handwriting, OCR correction, Polish

1 Introduction

Contemporary medical documents are created mostly in electronic form, but thousands of medical documents created and stored in archives are a possible source of very valuable knowledge. The task of off-line recognition of medical handwritten documents is distinguished from the general handwriting optical character recognition (OCR) by two important aspects. The task seems to be easier as documents come from a known source, e.g. a hospital and the domain and the group of authors are limited. On the other hand, medical handwriting is commonly recognised as an example of an almost illegible handwriting.

The difficult task of recognition on the level of letter images can be supported by the prediction of letter and word sequences on the level of language modelling. The problem of letter sequence modelling seems to be well investigated and solved in the case of the particular system discussed in the paper, e.g. [1]. However, on the level of word sequences the methods of morpho-syntactic modelling and stochastic language models did not give a fully satisfying solution. The former had too low accuracy, the latter were too sensitive for the differences between the learning and testing corpora. Moreover, both approaches do not utilise the semantic information for the OCR correction. The aim of this work is the construction of handwriting OCR correction algorithm utilising the information concerning the semantic properties of words. The method should not depend on any manually constructed semantic resource — all information should be directly extracted from a learning corpus of text.

* This work was financed by the Ministry of Education and Science, the project No 3 T11E 005 28, and partially by the Inst. of Applied Informatics WUT.

2 Semantic model for correction of OCR output

2.1 Task formulation

The main task of the constructed OCR system is to transform a handwritten medical document into its electronic version. We assume that documents come from one institution (e.g. a hospital ward), that limits their domain, and that they are written in Polish with some possible addition of foreign words. The whole system is divided into into three main parts (layers) [1]:

- a *character level* — soft recognition of isolated characters; for each character, a subject of recognition, the classifier fetches the vector of support factors for all characters in the alphabet,
- a *word level* — works on the results of character classification, isolated words are recognised using also soft classification paradigm: Hidden Markov Models (HMM) and Probabilistic Lexical Language Models (PLLM) combined with a classifier based on an incomplete probabilistic lexicon; the combined character level and word level are called a *word classifier* (WC),
- a *language modelling level* based on the rejection approach applying a *language model* (LM) to lists of *candidates* generated by WC, where a candidate is a possible recognition for a word position in the input text.

Our experiments were performed on the basis of *The Medical Corpus of the Institute of Applied Informatics* (KorMedIIS) [2] of electronic medical texts that was collected from the database of a hospital for which the prototype OCR system is being constructed. The collected texts belong to several categories but only The Corpus of Epicrises (CE) was used in the experiments. An epicrisis is short descriptions of a patient stay in a hospital, consists of several sentences (phrases), reports some details of the patient stay and treatment, and often copies after the other documents. CE includes 15 251 epicrises (1 701 313 words). During experiments, two testing parts of CE were randomly selected: TC1 including 300 epicrises (32 694 words) and TC2 with 2 240 epicrises (206 662).

As we were not able to collect a sufficiently numerous set of hand-written texts, a simulated experiment was performed. Text images were artificially created using the set of images of 5 080 hand-written characters, manually classified. A text image is created in the following steps. First, the text to be recognised is randomly drawn from the Test Corpus. Next, for each character one image of this character is randomly selected from the set of character samples. Finally, the drawn character images are arranged side by side into an artificial text image.

The simplified version of the WC used in experiments assumes that word images are correctly segmented into isolated characters. However we do not deal here with the non trivial problem of word segmentation, but we rather focus our attention on the efficient construction of a LM for the domain of medical texts written in an inflective language, e.g. Polish. For each written word in the text, the WC produces a list of the $k = 10$ most probable *candidate words* (henceforth *candidates*). Each candidate is assigned a *score*, i.e. a number which denotes how likely that word was correctly recognised according to the used classifier. In the

case of punctuations and numbers it is assumed that the WC produces perfect recognitions and returns only one candidate. The task of the LM is to select the proper recognition among the k possible for each position.

In [1], a morpho-syntactic LM was applied to lists of the k-best candidates independently of the WC scores. Next both classifiers were combined. The n-gram LMs of [3] were applied to the k-best lists, but WC scores were not used. The best candidate sequence was identified among the k-best ones. As the Viterbi search used in [3] produces the best path across the candidates, but gives no ranking of candidates on the subsequent positions, there is no simple way to combine the LMs of [3] with the scores of the WC. Contrary to this, we assumed here a gradual improvement of the result of the WC. In that way we want to omit the merging problems encountered in [3] and partially in [1], too. We want to explore WC scores in a style of the *grey-box* approach: not depending on the exact mechanisms of the WC, but taking into account its characteristics.

In order to make more space for the improvement introduced by the LM, we used in all experiments a version of the WC with the decreased accuracy. The WC achieves accuracy of 86.05% of words correctly recognised when the first candidate form the list is taken (96.69% in $k = 10$ candidates) as calculated for all *input tokens* from the TC1 corpus (86.09% and 96.37% respectively for TC2). The accuracy measured for *ambiguous tokens* (WC returned more than one candidate with the score above 0) is respectively: 80.2% and 95.31% ($k = 10$) for TC1 and 80.39% and 94.89% ($k = 10$) for TC2. We distinguished also *tokens without the answer* (TWA) among candidates. After testing the accuracy of the WC for the different values of k, we decided to identify a cut-off threshold τ for scores in order to balance the number of candidates analysed by the LM and the maximal possible accuracy. Two parameters were defined:

- a *cut-off precision* — the precision of recognition calculated only for ambiguous tokens, excluding TWAs, with at least one candidate above τ,
- a *coverage* — the number of candidates below τ in relation to all candidates from ambiguous tokens.

The cut-off precision determines the highest possible accuracy of decisions to be achieved by the LM. The coverage informs about the decrease in the number of candidates to be analysed by LM. The relation of the cut-off precision, coverage and threshold values for TC1 is presented in Fig 1 (for TC2 this relation is similiar). On this basis we selected $\tau = 0.85$, that corresponds to the cut-off precision $\approx 99\%$ and coverage $\approx 43.9\%$, i.e. less than half of the candidates are eliminated from tokens for which decission must be made.

2.2 Models

The idea is to select candidates, one per token, such that they form a group of the maximal average pair-wise *semantic similarity*. The measure of semantic similarity is extracted from the training corpus, see Sec. 2.3. In that way, a sequence of semantically related candidates is chosen as the recognition. Only

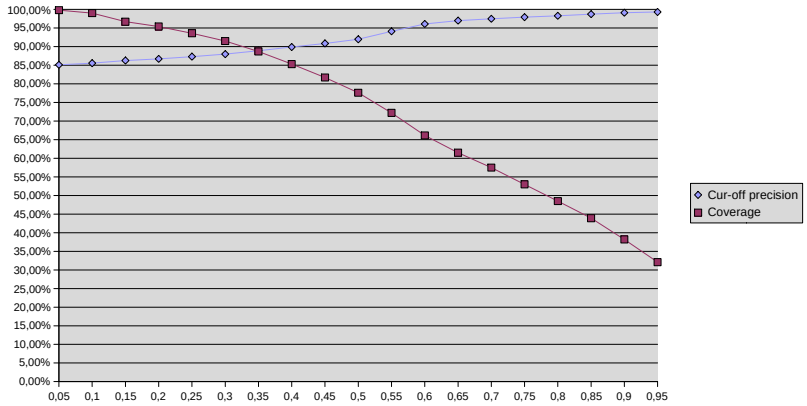

Fig. 1. The relation of the cut-off precision, coverage and threshold values.

candidates above the defined threshold τ are analysed. The general scheme of the algorithm is based on the *k–means algorithm* [4]:

– For each *text context* (a *document* or a *text window*):
  1. An initial centre of the cluster — a mean vector, is set.
  2. For each token a candidate which is the most similar to the mean vector is chosen.
  3. The mean vector is recalculated according to the chosen candidates.
  4. `If` a stop condition is not fulfilled `go to` 2.

Each candidate is represented by a vector of real values in the (possibly transformed) coincidence matrix (see Sec. 2.3). The initial mean vector is calculated in the step 1 as the average of the vectors of the first candidates of these tokens, which are unambiguous after the τ elimination. Moreover, because of the numerous *ad hoc* abbreviations and symbols, unambiguous tokens of less than three letters, e.g. "*ż.*" — the *ad hoc* abbreviation of "życia" ($life_{case=gen}$) are not taken into account during the initial centre calculation. If there is no unambiguous token, one token is randomly chosen and its best candidate is taken as the initial mean. If a text window is used as a context, then for each next position of the window, the initial mean vector is a sum of the average with the previous mean vector — we try to keep consistency across the whole document.

For each ambiguous token we select a candidate which is the most *similar* to the mean vector. Different measures of similarity were tested, see Sec. 2.3. Next, the mean vector is modified to the average of all selected candidates. The process is repeated until some number of iterations is reached or the increase of the average similarity in relation to the best level so far is below some threshold.

After the first experiments, see Sec. 3, we noticed that smaller text windows result in better accuracy. Moreover, the very positive results of the application

of word n-gram LMs to this task [3] show that the short distance associations among tokens are very important here. Thus, we proposed a new LM based on semantic similarity called the *Semantic Window* LM (SemWnd). In SemWnd, a *flexible* text window of the minimal size N is moved across a document in left to right direction. The centre of the window is positioned on ambiguous tokens. Next, the borders of the window are gradually extended to the left and right until $N/2$ *content tokens* are collected on both sides. A content token is an input token that can possibly be a good discriminator. Because of the similarity measures used (similarity is calculated for every pair of tokens) here we define content token as a token that has the best candidate of more than 3 letters.

The score of a candidate c_i of the ambiguous token in the centre of the text window is calculated according to SemWnd as following:

$$score(c_i)_{SW} = \sum_{N \geq j \geq 1} sim(c_i, best\_of(t_j)) \tag{1}$$

where t_j is a token in the window, and $best\_of$ returns its best candidate.

The best candidates in the left part of the window, in the case of ambiguous tokens, are defined according to the value of $score()_{SW}$, in the right part according to the scores of the WC. In the case of odd values of N, we take the left part bigger, as in the opposite case the results of the experiments were lower. The accuracy of SemWnd is higher than the accuracy of the WC alone.

2.3 Semantic similarity for words

As there is no thesaurus covering the terminology of KorMedIIS, the used *Semantic Similarity Measures* (SSMs) are extracted directly from the corpus. According to the *Distributional Hypothesis* of Harris [5] words occurring in similar contexts have similar meanings. If we are interested only in SSM, then the best results can be achieved when contexts are described by morpho-syntactic features of word occurrences, e.g. [6]. However, in the case of the OCR correction, we need a SSM working for all tokens occurring on the input, including non-words and any symbols. They have no morpho-syntactic properties. Moreover, we want to predict occurrences of particular word forms as following other word forms. Thus, we decided to use tokens (word forms and symbols) as elements being described and to identify the context with co-occurrence of a described token t_i with some other token t_j in a short text window of 5-15 tokens. A *coincidence matrix* of the form $\mathbf{M}[t_i, t_j]$ is created from the learning part of KorMedIIS, where t_i,t_j are two tokens, and the cell contains the number of co-occurrences of t_i and t_j in the same text window with t_i is in its centre. The rows of $\mathbf{M}$ correspond to content tokens (see Sec. 2.2) occurring in the learning corpus, while the columns initially corresponded to all tokens, including non-content ones, too. After the first experiments, we limited the column tokens (i.e. features) only to the ones occurring at least min_{tf} times in the corpus with $min_{tf} = 3$ defined experimentally. Elimination of columns for non-content tokens resulted in the decreased accuracy of the model as non-content tokens express some description

of contexts. We did not apply any weight function to the frequencies to preserve the possibility of the probabilistic interpretation of the data.

According to collected earlier experience [7, 6], we tested three possible SSMs:

- the *cosine measure* (Cos) applied to row vectors transformed previously by the *logarithmic scaling* and *entropy normalisation* [7],
- *Information Radius* ($IRad$): $D(p||\frac{p+q}{2})+D(q||\frac{p+q}{2})$, and similarity: $Sim_{IRad} = 10^{-\beta IRad(p||q)}$, where p and q are the probability distributions calculated for the two row vectors being compared, $D(p||q) = \sum p \log \frac{p}{q}$ is Kullback-Leiber divergence, $p(t_i, t_j) = \frac{\mathbf{M}[t_i,t_j]}{TF(t_i)}$, $TF(t_i)$ — the total frequency of t_i in the learning corpus [8], we choose $\beta = 10$ like in [7].
- and the cosine measure applied to vectors produced by the *Rank Weight Function* (RWF) proposed in [6], discussed shortly below.

The aim of RWF is to identify a set of the most descriptive features (matrix columns) for a given row vector $\mathbf{M}[t_i, \bullet]$ (word), and to describe $\mathbf{M}[t_i, \bullet]$ with the ranking of the features in place of the exact feature values [6]:

1. Weighted values of the cells are recalculated using a weight function f_w: $\forall_{t_j}\mathbf{M}[t_i, t_j] = f_w(\mathbf{M}[t_i, t_j])$.
2. Features in a row vector $\mathbf{M}[t_i, \bullet]$ are sorted in the ascending order on the weighted values.
3. The k highest-ranking features are selected; e.g. $k = 1000$ works well.
4. For each selected feature t_j: $\mathbf{M}[t_i, t_j] = k - rank(t_j)$.

Following [6], as the weight function f_w, we applied the *t–score* measure of statistical significance of the pair frequency: t_i and t_j.

2.4 Heuristic rules

Documents of KorMedIIS as written during the real practice in a hospital, are full of short phrases, *ad hoc* abbreviations, and lists of activities or medicines. The 'non-standard' parts are especially difficult for the LMs. Thus we introduced *heuristic rules* correcting some specific cases which were manually identified in the learning data. Two types of rules were defined: *simple rules* and *morpho-syntactic rules*. Simple rules deals with mistakenly recognised short symbols and repair some constant errors made by the WC, e.g. a rule choosing 'x' in the case of the ambiguity between 'x' and 'k' (the WC prefers 'k', what is wrong in the vast majority of cases). Several simple rules were defined.

Morpho-syntactic rules refer to the morpho-syntacic features of candidates which are obtained from the morphological analyser *Morfuesz* [9]. In the case of Polish, an inflective language, the features constrain the possible sequences of candidates. The rules express constraints that should be preserved by proper candidate sequences. Candidates fulfilling the constraints are preferred. Only several rules were constructed so far, as most candidates are ambiguous according to their morpho-syntactic description (an intrinsic property of the inflective language) and the documents contains quite numerous grammatical and spelling

errors. The examples of rules are: the rule testing presence a case required by a preposition or the rule testing the possibility of the morpho-syntactic agreement between a noun and an adjective which precedes it.

The rules are applied during selection of candidates for a token in the centre of the text window. From the candidates fulfilling the constraint the one with the highest $score_{SW}$ (1) is chosen. In the case there is no candidate matching, all are evaluated by the SMM. All the rules, simple and morpho-syntactic, when applied increase the accuracy by 0.5%.

2.5 Merging with the Word Classifier

The results achieved by SemWnd (see Sec. 3) alone, e.g. 92.69% for the TC1, are comparable with the n-gram model (92.8% for the same fold). However, during the manual inspection of the errors we discovered that in the case of the most serious mistakes of SemWnd, e.g. a candidate unrelated to the context selected or a typo introduced (because of typos present in the CE) the WC scores are mostly opposite to the scores of SemWnd. The both classifiers, which are working on different levels and on the basis of different data, seem to be often complementary in their decisions. Thus, we decided to combine the scores and to use a WC score (from $\langle 0, 1 \rangle$) as a scaling factor: $score(c_i)_{WC+SW} = score(c_i)_{SW} * score(c_i)_{SW}$. This simple merge increased the accuracy by about 2%–3%.

3 Experiments

In order to directly compare our present approach with the previous works, namely [1, 3], we used the same learning and test corpus — TC1. The best result obtained for the TC1 part used in [3] is 92.80%. Two basic parameters were tested: the size of context during the construction of coincidence matrix and the size of the text window used during selection of candidates for ambiguous tokens. For the construction phase the best result was achieved for context of 10 tokens.

Moreover, we tested two methods: the k-means model as the first one and SemWnd. In both models the best results, all presented in Tab. 1, were obtained by applying SMM based on RWF. During the first experiments with SemWnd only simple rules were used, in the next three rules of both types were applied

The morpho-syntactic LM of [1] (with some heuristic rules) produced 89.02% accuracy on the same corpus. It means that the best results obtained with the help of SemWnd(10,3), i.e. 93.1% (the reduction of the recognition error by 59.7%) outperforms this result. It is also higher that the result of the n-gram LM equal to 93%. However, the difference is not statistically significant. On the larger corpus, i.e. the whole TC, the accuracy of the n-gram model is 92.8%, while the accuracy of the SemWnd(10,3) is 92.69%, but this difference is not statistically significant, too. After merging with WC, SemWnd achieved 94.64%. On the TC2 corpus the accuracy of the n-gram LM is 91.69%. SemWnd achived lower accuracy of 90,88%, but after merging its accuracy increased to 94.28%.

The other SMMs produced significantly lower accuracy in SemWnd. On the TC1 corpus, the best accuracy of the cosine SMM was 91.3%, and of the IRad was 90.9%. This observation is consistent with the results obtained in [7, 6].

Method	Corpus	Window size				
		Correction				
		3	5	10	15	document
k-means	TC1(2000)	89.45	91.15	90.65	90.80	88.60
SemWnd+rules	TC1(2000)	93.10	92.80	92.05	91.95	—
SemWnd+rules	TC1	92.69	92.51	91.74	91.49	—
SemWnd+rules	TC2	90.88	90.75	89.82	89.36	—
WC+SemWnd+rules	TC1(2000)	95.10	95.00	95.00	95.00	—
WC+SemWnd+rules	TC1	94.64	94.54	94.37	94.26	—
WC+SemWnd+rules	TC2	94.28	94.21	94.02	93.97	—

Table 1. The overall accuracy [%] of the models (TC1,TC2 — folds of the corpus, TC1(2000) — the first 2000 tokens of TC1).

4 Related work

It is very hard to find applications of SSMs to the OCR correction in literature. Most works concern applications in Speech Recognition (SR) or correction of spelling errors. In [10] a SSM extracted by the LSA technique [11] (e.g. using the cosine measure) is applied in SR. This approach depends on the similarity threshold set manually for the given domain. Moreover, LSA limits the number tokens processed according to high memory complexity. A SSM based on the retrieval of co-occurrence frequencies of words in a kind of encyclopedia is proposed in [12]. In [13], only a limited subset of "content words" is processed in SR. A pair-wise similarity (PMI measure applied) of words is calculated in a way similar to [14] based on the manually constructed Roget thesaurus. In [14] a SSM based on the counting the number of relation defined between two words in the Roget thesaurus is used in OCR correction. In that way the possibility of correction is limited only to the words present in the thesaurus. The reported overall accuracy is higher than ours, but the method was applied to printed OCR of texts written in a 'standard language'. Lexico-semantic patterns together with a lexicon are applied in detection and correction of errors in a kind of SR system in [15]. However, the method strongly depends on the costly resources: an ontology and a lexicon with semantic classes assigned to lexemes. In [16], the notion of "a low density language" is introduced, i.e. a language with a limited electronic resources. The sub-language of KorMedIIS can be treated as a low density language, too. They argue that for such languages, the methods working on the level of letter models are better suited. In [17] a notion of lexical chains based on semantic similarity is used in the detection and correction of spelling errors.

All words which occurr less times than some threshold and are not similar to any other words are identified as suspect. Next, the different spelling variants of suspect words are tested. The used SSM is based on a thesaurus and the whole approach is limited only to nouns. [18] is another work in the area of the spelling correction and the context sensitive spelling error detection. An approach based on LSA is proposed. The method works exclusively on "confusion sets" i.e. the sets of the most frequent spelling errors. In the similar [19] the learning process is performed on the basis of errors artificially introduced to documents. The errors are generated according to a manually defined set of rules

5 Conclusions

SemWnd outperforms the morpho-syntactic LM proposed in [7]. The documents of KorMedIIS are very specific. They were written on a computer and often one document contains parts copied from the others. The percentage of typos and *ad hoc* created abbreviations is very high (more than 25% of tokens are not proper words). Thus, the problem is positively biased for the applied n-gram model with a simple Laplace smoothing (almost 'memory-based'). However the merge of SemWnd with the WC produced significantly better results. On the other hand, as we tested the gradual decrease of the learning corpus size, we could observe the lower speed of the accuracy decrease in the case SemWnd model, in comparison to the n-gram model. SemWnd expresses the better ability to generalise.

SemWnd model is more flexible in combing it with other classifier in comparison to the n-gram LM. The n-gram LM of [3] produces only the best path across the candidates. It does not return probabilities or scores for candidates. If we want to use that model for the reduction of the candidate lists, we have change the search algorithm to the full search in place of the Viterbi-like search. The time complexity becomes unacceptable. In the case SemWnd there is no such problem — its time complexity is independent on the number of candidates left.

The SSM in SemWnd tends to group different word forms of the same lexeme as semantically similar. Unfortunately, different misspelled variants of a word form are grouped together with it, too. Merging with the WC helps here, but a more general approach should be developed. To increase the accuracy of SemWnd, we need to combine it with a more sophisticated morpho-syntactic analysis of candidate sequences and to develop some spelling correction.

References

1. Godlewski, G., Piasecki, M., Sas, J.: Application of syntactic properties to three-level recognition of Polish hand-written medical texts. In Bulterman, D., Brailsford, D.F., eds.: Proc. of the 2005 ACM Symposium on Document Engineering, New York, ACM Press (2006)
2. Piasecki, M., Godlewski, G., Pejcz, J.: Corpus of medical texts and tools. In: Proceedings of Medical Informatics and Technologies 2006, Silesian University of Technology (2006) 281–286

3. Piasecki, M., Godlewski, G.: Language modelling for the needs of OCR of medical texts. In Maglaveras, N. et. al., eds.: Biological and Medical Data Analysis. 7th International Symposium, ISBMDA 2006, Thessaloniki, Greece, December 7-8 2006. LNCS, Springer (2006)
4. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (2001)
5. Harris, Z.S.: Mathematical Structures of Language. Interscience Publishers, New York (1968)
6. Piasecki, M., Szpakowicz, S., Broda, B.: Automatic selection of heterogeneous syntactic features in semantic similarity of Polish nouns. In: Proceedings of the Text, Speech and Dialog 2007 Conference. LNAI 4629, Springer (2006)
7. Piasecki, M., Broda, B.: Semantic similarity measure of Polish nouns based on linguistic features. In Abramowicz, W., ed.: Business Information Systems 10th International Conference, BIS 2007, Poznań, Poland, April 25-27, 2007, Proceedings. LNCS 4439, Springer (2007)
8. Dagan, I., Lee, L., Pereira., F.: Similarity-based method for word sense disambiguation. In: Proc. of the 35th Annual Meeting of the ACL, Madrid, Spain, ACL (1997) 56–63
9. Woliński, M.: Morfeusz — a practical tool for the morphological analysis of Polish. [20] 511–520
10. Cox, S., Dasmahapatra, S.: High-level approaches to confidence estimation in speech recognition. Speech and Audio Processing, IEEE Transactions on **10**(7) (2002) 460–471
11. Landauer, T., Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition. Psychological Review **104**(2) (1997) 211–240
12. Kupiec, J., Kimber, D., Balasubramanian, V.: Speech-based retrieval using semantic co-occurrence filtering. Proceedings of the workshop on Human Language Technology (1994) 373–377
13. Inkpen, D., Désilets, A.: Semantic Similarity for Detecting Recognition Errors in Automatic Speech Transcripts. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (2005) 49–56
14. Jobbins, A., Raza, G., Evett, L., Sherkat, N.: Postprocessing for OCR: Correcting Errors Using Semantic Relations. Language Engineering for Document Analysis and Recognition (LEDAR), AISB96 Workshop, Sussex, England (1996)
15. Jeong, M., Kim, B., Lee, G.: Semantic-Oriented Error Correction for Spoken Query Processing. 2003 IEEE Workshop on Automatic Speech Recognition and Understanding (2003) 156–161
16. Kolak, O., Byrne, W., Resnik, P.: A generative probabilistic OCR model for NLP applications. Proc. of the 2003 Conf. of the North American Chapter of the ACL on Human Language Technology-Volume 1 (2003) 55–62
17. Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Natural Language Engineering **11**(01) (2005) 87–111
18. Jones, M., Martin, J.: Contextual spelling correction using latent semantic analysis. Proc. of the 5th Conf. on Applied Natural Language Processing (1997) 166–173
19. Al-Mubaid, H., Truemper, K.: Learning to Find Context-Based Spelling Errors. Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques (2006)
20. Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K., eds.: Intelligent Information Processing and Web Mining — Proc. of the International IIS: IIPWM'06, Zakopane, Poland, June, 2006. Advances in Soft Computing. Springer, Berlin (2006)